2024-2-19

Manuscript on "Improving Nonlinear Dynamical Probabilistic Latent Variable Model for Inferential Sensors"

Zhichao Chen[1], Hao Wang[1], Guofei Chen[1], Yiran Ma[1], Le Yao[2], Zhiqiang Ge[3] and Zhihuan Song[1]
[1]Zhejiang University, [2]Hangzhou Normal University, [3]Pengcheng Laboratory

Keywords: Latent Variable Model, Machine Learning, Inferential Sensors, Bayeisan Inference

1. Introduction

Despite advancements in network architectures for feature extraction, applying NDPLVMs to inferential sensor tasks raises two critical yet overlooked questions. We define key elements for clarity: unobservable latent variable (z), covariates (u), label (y), inference network q, generative network p and the transition function (f). The NDPLVMs training algorithm employs Amortized Variational Inference (AVI), which utilizes two neural networks: the inference network (q) to deduce latent variable z from data (u), and the generative network (p) to decode label y from z. This approach aims to minimize two primary components: the regularization term, indicating the deviation between inferred and prior latent spaces, and the likelihood term, the difference between original and generated data, both quantified using probabilistic density functions. The essential challenges in this process are summarized as follows:

1. **Inaccurate Inference of Latent Space:** According to the Bayes' theorem, the inference network's inputs should align with the structure of the generative network $p(y|z)$ / $p(u, y|z)$. In the context of inferential sensor tasks, where the generative network invariably involves y, the inference network should ideally be formulated as $q(z|y)$ / $q(z|u, y)$. However, most of current works have not incorporated the label information y into the inference network. This omission could limit model performance due to inaccurate inference of latent variable z. The key to addressing this issue is reframing the model learning problem into an optimization problem and selecting the inference network's input based on solving the optimization problem.
2. **Model Implementation within DL Backends:** It is imperative to note that the foundational framework of NDPLVMs derivation is predicated on the probabilistic density functions (pdf) form. However, DL backends [1] are fundamentally structured around individual data samples, which are instances sampled from the pdf. This divergence between the theoretical pdf and practical data samples engenders difficulties in computational realization. To delineate further, consider f tasked with mapping a latent variable z over a time increment from t to $t+1$ ($z_{t+1} = f(z_t)$). Drawing upon the celebrated Liouville's theorem, the pdf must adhere to the equation $p(z_{t+1}) \det \partial_z f(z_t) = p(z_t)$, which necessitates the computation of the Jacobian matrix. This requirement, unfortunately, gives rise to a substantial realization in network implementation by DL backends. Consequently, a pivotal aspect is the derivation of moment expressions like mean and covariance, a step vital in bridging the gap between pdf and DL backends.

On this basis, this paper addresses the identified challenges by introducing a new NDPLVM model, termed Optimal Control-NDPLVM (OC-NDPLVM), tailored for inferential sensor tasks. Specifically, we derive its learning objective using stochastic differential equation theory, examine its parameter learning based on the celebrated Alternating Direction Multiplier Method (ADMM), and rigorously analyze its architectural components. In this analysis process, our approach reveals that the inference network mirrors an optimal control (OC) subproblem in ADMM, providing the analysis of solution property of OC problem to guide the selection of inference network's input and therefore address issue 1). To address issue 2), we meticulously analyze the moment expressions within the neural network structure. Furthermore, we summarize the training& testing inference algorithm and discuss its convergence properties. Finally, empirical validation is provided through experiments on two industrial process datasets, demonstrating the efficacy of our approach. The paper's contributions are summarized as follows:

- We propose the a novel NDPLVM model named OC-NDPLVM and derive its loss function from SDE theory rigorously.
- We reconceptualize the parameter learning of the OC-NDPLVM as an optimization problem. By solving this optimization problem, we redesign the inference network's input.
- We derive an approximation of the moment expressions in OC-NDPLVM for numerical implementation.

2. Related Works (Will Be Released After Acceptance)

2.1. NDPLVMs for Inferential Sensor

2.2. Inference of Latent Space

2.3. Model Implementation in DL Backends

2.4. Technical Gaps Summary

In summary, our research addresses two critical, yet previously neglected issues:

1. Determining the appropriate inputs for the inference network and infer the latent space more accurately.
2. Developing methods to derive moment expressions, such as mean and covariance, for model implementation.

These challenges have not received adequate attention in prior studies. To bridge these gaps, we introduce the OC-NDPLVM, its associated parameter learning algorithm, and the formulation of moment expressions in this paper.

3. Preliminaries

3.1. Amortized Variational Inference

Let y and z be the observed and latent variables, respectively. Variational inference tends to approximate the posterior distribution of the latent variable $p(z|y)$ with the variational distribution $q(z)$ by minimizing their Kullback-Leiber divergence (KL divergence), which can be refromulated as the maximization of the Evidence Lower BOund (ELBO) for model training [2]:

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}(q(z)\|p(z|y)) &= \int q(z)\log\frac{q(z)}{p(z|y)}\mathrm{d}z \\ &= \overbrace{\int q(z)[\log\frac{q(z)}{p(z)} - \log p(y|z)]\mathrm{d}z}^{\text{ELBO}} + \log p(y).\end{aligned} \tag{1}$$

From the derivation presented in (1), it becomes apparent that our learning objective evolves to be the ELBO, given that $\log p(y)$ retains its status as a constant.

Note that, the optimal $q^*(z)$ is approximate to $p(z|y)$. Built upon this, to estimate the optimal variational distribution $q(z)$, AVI employs a stochastic function $q_\phi(z|y)$ that maps the observed variable to the latent variable belonging to the variational posterior density; the parameter ϕ is learned during the optimization process [3]. Moreover, in the context of AVI, it is a common assumption that $q_\phi(z)$ can effectively model the optimal variational distribution $q^*(z)$. If we consider the optimal variational distribution $q^*(z)$ as a function, the fundamental goal of AVI is to identify the input variable of this function and approximate it using a function parameterized by ϕ. In this way, the model can infer the latent variables for new data points, without re-running the optimization process.

3.2. Stochastic Differential Equation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ [4] be a probabilistic space, where Ω is the sample space, σ-algebra $\mathcal{F}$ is the set of events, and $\mathbb{P}$ is probability measure: $\mathcal{F} \mapsto [0,1]$. Let W_t be a $\mathcal{F}_t$-adapted Wiener process on this probabilistic space, $b(x,t)$ and $L(t)$ be two $\mathcal{F}_t$-adapted stochastic process, we have an Itô process:

$$x(t) = x(0) + \int_0^t b(x,\tau)\mathrm{d}\tau + \int_0^t L\mathrm{d}W_t, \tag{2}$$

which is the solution to the stochastic differential equation:

$$\mathrm{d}x(t) = b(x,t)\mathrm{d}t + L\mathrm{d}W_t, \tag{3}$$

where $b(x,t)$ is referred to the drift term, L is referred to the volatility term.

Based on the above mentioned concepts, the the likelihood ratio of two Itô processes is given as follow based on the celebrated Girsanov theorem and Radon-Nikodym theorem [5]:

Theorem 3.1 (Likelihood ratio of Itô Process)**.** *Based on (3), we can introduce two Itô processes as follow:*

$$\begin{aligned}\mathrm{d}x &= f(x,t)\mathrm{d}t + \mathrm{d}W, x(0) = x_0, \\ \mathrm{d}y &= g(y,t)\mathrm{d}t + \mathrm{d}W, y(0) = x_0,\end{aligned} \tag{4}$$

where $f(x,t)$ and $g(y,t)$ are drift terms. The Radon-Nikodym derivative along their respective path measures $\mathbb{P}$ and $\mathbb{Q}$ is given by:

$$\begin{aligned}\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x) = &\exp(-\frac{1}{2}\int_0^t \|g(x,\tau) - f(x,\tau)\|^2\mathrm{d}\tau \\ &+ \int_0^t (g(x,\tau) - f(x,\tau))^\top \mathrm{d}W).\end{aligned} \tag{5}$$

From this theorem, we can further obtain the KL divergence of two Itô processes as follow [4, 5] (the detailed derivation is given in Section **??** of Supplementary Material):

$$\mathbb{D}_{\mathrm{KL}}(\mathbb{Q}_t \| \mathbb{P}_t) = \mathbb{E}_{\mathbb{Q}(z)}[\frac{1}{2}\int_0^t \|\nu\|^2 \mathrm{d}\tau], \tag{6}$$

where ν is defined as the follow equation according to (5):

$$L\nu = g(x,\tau) - f(x,\tau), \tag{7}$$

and $\mathbb{E}$ is expected operator.

Note that, compared to conventional KL divergence between two d-dimensional Gaussian distribution (denoted as $q(z) \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $p(z) \sim \mathcal{N}(\mu_2, \Sigma_2)$), which widely applied in conventional NDPLVMs derivation:

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}(q(z)\|p(z)) = \frac{1}{2}[\log\frac{\det\Sigma_1}{\det\Sigma_2} + \mathrm{Tr}\left(\Sigma_1^{-1}\Sigma_2\right) \\ + (\mu_1-\mu_2)^\top\Sigma_1^{-1}(\mu_1-\mu_2) - d].\end{aligned} \tag{8}$$

The KL divergence between two Itô processes, as defined in (8), adopts a quadratic form that eliminates the need for matrix inversion computations. This simplification greatly facilitates the process of model derivation. Moreover, the Itô process is closely linked to Brownian motion, a phenomenon frequently observed in various natural occurrences. It is worth emphasizing that industrial processes commonly employ the Kalman Filter, which can be viewed as a variant of the Itô process [5]. Taking these factors into consideration, we have chosen the Itô process as the model prior in this paper.

3.3. ADMM Algorihtm

Consider a special case where the decision variables w and v in the objective function term are separable (assume f and h are convex):

$$\begin{aligned} \min \quad & \mathrm{Obj}(v,w) = f(w) + h(v) \\ s.t. \quad & Aw + Bv = c \end{aligned} \quad . \tag{9}$$

According to the augmented Lagrangian multiplier method [6], we can cast (9) to an unconstrained optimization problem, with the objective function:

$$L = f(w) + h(v) + \lambda^\top(Aw + Bv - c) \tag{10}$$

where λ is Lagrangian multiplier, and ρ is quadratic penalty coefficient. The celebrated ADMM algorithm [6] is a common algorithm to solve (10), where variables u and w are optimized separately at each iteration:

$$\begin{cases} w^{k+1} & = \arg\min_w L(w, v^k, y^k) \\ v^{k+1} & = \arg\min_v L(w^{k+1}, v, y^k) \end{cases} . \tag{11}$$

4. Proposed Approach

4.1. Problem Statement

In this study, our focus is solely on inferential sensor tasks under a supervised learning context. This domain represents a specialized subset of time-series analysis. Extending the concepts presented in reference [7], we define the forecast horizon as H and the historical sequence length as T. Based on this framework, our task is as follows: Given the historical sequence of key indices $y_{1:\mathrm{T}} \in \mathbb{R}^{\mathrm{T}}$ and the sequence of covariates $u_{1:\mathrm{T+H}} \in \mathbb{R}^{\mathrm{D}\times(\mathrm{T+H})}$, we aim to predict the key index for the next H steps, specifically $y_{\mathrm{T+1:T+H}} \in \mathbb{R}^{\mathrm{H}}$.

4.2. Learninig Objective Derivation

Based on the Section 4.1, the learning objective can is defined as follow based on the maximum likelihood estimation:

$$\arg\max \quad \log p(\vec{y}|\vec{u}) = \arg\max \quad \log \int p(\vec{y}, \vec{z}|\vec{u})\mathrm{d}z. \tag{12}$$

However, the right-hand-side of (12) is intractable. To solve this problem, the assumption on latent space z is introduced.

As introduced in section 3.2, compared to simpler random walks, the Itô process accommodates a higher degree of complexity and variability in modeling dynamic systems. Moreover, the Itô process, when utilized as a prior, lends itself to the direct application of established results and theorems from the theory of stochastic processes. Lastly, in Bayesian inference, a critical task is to update our knowledge about unknown parameters as new data becomes available.

Therefore, based on Sections 3.1 and 3.2, we first define the prior Itô process to describe the transition between states at different time with parameter θ as follow:

$$\mathrm{d}z = f_\theta(z, u)\mathrm{d}t + L\mathrm{d}W_t. \tag{13}$$

To align with the prior Itô process, the posterior is designed as another Itô process with parameter ϕ as follows to approximate it:

$$\mathrm{d}z = f_\phi(z, u)\mathrm{d}t + L\mathrm{d}W_t. \tag{14}$$

Besides, we define ν as follow:

$$L\nu = f_\phi(z, u) - f_\theta(z, u). \tag{15}$$

Based on (14) and (15), the inference network q is converted to the control policy ν.

Consolidating (12) to (15), we can obtain the following proposition of our model learning objective:

Proposition 4.1. *Optimizing* (12) *is equivalent to optimize the problem defined as follow:*

$$\begin{aligned}
\min_{\theta,\nu} \quad & \sum_{t=1}^{\mathrm{T}} \{\mathbb{E}_{\mathbb{Q}(z)}[-\log p_\theta(y_t|z_t, u_{1:t}) + \int_{t-1}^{t} \frac{1}{2}\|\nu\|^2 \mathrm{d}\tau]\} \\
s.t. \quad \mathrm{d}z \quad & = f_\phi(z, u)\mathrm{d}t + L\mathrm{d}W_t \\
& = f_\theta(z, u)\mathrm{d}t + L\nu\mathrm{d}t + L\mathrm{d}W_t
\end{aligned} \tag{16}$$

Proof. The proof is given in Section **??** of Supplementary Material. □

In Proposition 4.1, we reformulate the parameter learning optimization problem associated with NDPLVMs. This reformulation provides a perspective for understanding the core principles underlying parameter learning within convex optimization framework. However, the learning objective outlined in (16) proves to be challenging to optimize owing to the indeterminate initial points across various intervals. Additionally, its computation necessitates a “backtracking" operation, an impractical approach for inferential sensor tasks given their inherent causality. Fortunately, this problem can be solved based on “one-step lookahead minimization" method according to reference [8]. And thus, we propose the following proposition to derive an upper bound for (16):

Proposition 4.2. *The objective function defined in* (12) *have the following upper bound:*

$$\begin{aligned}&\min_{\theta,\nu}\sum_{t=1}^{\mathrm{T}}\{\mathbb{E}_{\mathbb{Q}(z)}[-\log p_\theta(y_t|z_t,u_{1:t})+\int_{t-1}^{t}\frac{1}{2}\|\nu\|^2\mathrm{d}\tau]\}\\&\leq\\&\sum_{t=1}^{\mathrm{T}}\min_{\theta,\nu}\{\mathbb{E}_{\mathbb{Q}(z)}[-\log p_\theta(y_t|z_t,u_{1:t})+\int_{t-1}^{t}\frac{1}{2}\|\nu\|^2\mathrm{d}\tau]\}\end{aligned}\tag{17}$$

Proof. The proof is given in Section **??** of Supplementary material. □

4.3. Inference Network's Input Selection

Based on (17) in section 4.2, we can conduct concerning analysis: the right-hand-side of (17) two sets of variables to be optimized, viz. the model parameter θ and the control policy ν. It should be pointed out that this problem involves an integral term $\int$ with respect to ν in time domain, which belongs to an optimal control problem as per reference [9]. Based on this, we named our model OC-NDPLVMs. Moreover, it should be pointed out that the optimal control problem can be regarded as an infinite-dimensional optimization problem [10].

Comparing (17) with (9), we observe that the θ in (17) corresponds to the w in (9); the ν in (17) corresponds to the v in (9). The log-likelihood term $\log p_\theta(y_t|z_t,u_{1:t})$ and control policy terms $\int_{t-1}^{t}\|\nu\|^2\mathrm{d}\tau$ are separable. This observation motivates us to solve the parameter learning of NDPLVMs given in (17) through the lens of ADMM. Since the model parameter θ can be optimized through auto-differential backends such as PyTorch [1] by stochastic gradient descent-based optimizers like Adam, the minimization sub-problem concerns with θ will not be discussed. The rest of this subsection will discuss the minimization sub-problem (the OC problem) concerns with ν.

Based on Proposition 4.2, optimizing the *global* ELBO defined in (16) can be converted to optimizing the *local* ELBO within interval $[t,t+1]$ for $t\in[0,\mathrm{T})$. And thus, the following subsection will focus on the optimization of *local* ELBO within interval $[t,t+1]$. We will take the optimal control signal derivation between interval $[t,t+1]$ as an example to illustrate this sub-problem.
Based on the gaussian assumption on the observation data, the ELBO between interval $[t,t+1]$ can be expanded as:

$$\begin{aligned}&\mathbb{E}_{\mathbb{Q}(z)}[-\log p_\theta(y_t|z_t,u_{1:t})+\int_{t-1}^{t}\frac{1}{2}\|\nu\|^2\mathrm{d}\tau]\\=&\mathbb{E}_{\mathbb{Q}(z)}[\frac{1}{2}(y_t-\mu_t^y)^\top(y_t-\mu_t^y)+\int_{t-1}^{t}\frac{1}{2}\|\nu\|^2\mathrm{d}\tau],\end{aligned}\tag{18}$$

where we parameterize $p(y|z_t,u_{1:t})$ as $\mathcal{N}(\mu_t^y,\mathcal{I})$ as per references [11, 12]. Base on this setting, we conduct the approximation in the last line. Besides, the mean value μ_t^y is obtained via neural network denoted as g_θ:

$$\mu_t^y=\mathbb{E}[g_\theta(z_t)]\approx g_\theta(\mu_t^z).\tag{19}$$

Based on the approximation operations, the following optimal control problem can be formulated to obtain the optimal ν (denoted as ν^*):

$$\begin{aligned}\min_{\nu}\quad&(y_t-\mu_t^y)^\top(y_t-\mu_t^y)+\int_{t-1}^{t}\frac{1}{2}\|\nu\|^2\mathrm{d}\tau\\ s.t.\quad&\mathrm{d}z=f_\theta(z,u)\mathrm{d}t+L\nu\mathrm{d}t\end{aligned},\tag{20}$$

where the diffusion term is omitted since the model training mainly concentrates on the mean. And thus, the following Hamiltonian equation can be derived to solve the constrained optimal control problem according to the Pontryagin's maximum principle [9] :

$$\mathcal{H} = \frac{1}{2}\|\nu\|^2 + \lambda^\top (f_\theta(z,u) + L\nu), \tag{21}$$

where λ is Lagrangian multiplier. According to the sufficient condition for optimal control extreme value, the following equation can be obtained:

$$\frac{\partial \mathcal{H}}{\partial \nu} = \nu + L\lambda = 0. \tag{22}$$

Note that, the second-order derivative of the Hamiltonian function is a positive-definite matrix (identity matrix $\mathcal{I}$): $\nabla_\nu^2 \mathcal{H} = \mathcal{I} \succ 0$, which indicates that the generalized Legendre-Clebsch necessary condition can be satisfied [9]. As such, the extreme value obtained by the optimal control signal ν^* according to (21) is the minimum value.

According to the optimal control principle, the mean μ_t^z, and the co-state λ satisfy the following differential equations:

$$\begin{cases} \frac{\mathrm{d}\mu_t^z}{\mathrm{d}t} = \mathbb{E}(\frac{\partial \mathcal{H}}{\partial \lambda}) = f_\theta(\mu_t^z, u) + L\nu \\ \frac{\mathrm{d}\lambda}{\mathrm{d}t} = -\frac{\partial \mathcal{H}}{\partial z} = -\frac{\partial f_\theta(z,u)}{\partial z} \end{cases}. \tag{23}$$

The corresponding boundary condition for the equations are given as follow:

$$\begin{cases} \mu_{t-1}^z = \mathbb{E}(z_{t-1}) \\ \lambda_t = 2(\mu_t^y - y_t)^\top \frac{\partial g(z_t)}{\partial z_t}|_{\mu_t^z} \end{cases}. \tag{24}$$

By observing (21) to (24), the optimal ν mainly concerned with the observation data $y^\top$ at final time t, stochastic variable z_t (parameterized via mean μ_t^z and covariance Σ_t). It should be pointed out that, the computation of the Jacobian will result in a higher model training time. However, we know the optimal control is the function of y_t, μ_t^z, and Σ_t^z. On this basis, revist (15), we can simulate a neural network denoted as q with parameter ϕ as Section 3.1 mentioned:

$$\nu^* = q_\phi(y_t, z_t) = q_\phi(y_t, \mu_t, \Sigma_t). \tag{25}$$

Drawing on the solution of the Optimal Control (OC) subproblem delineated in (25), we have effectively addressed the issue of "inaccurate inference of latent space" by carefully selecting the input for the inference network. Moreover, our analysis extends beyond the conventional scope of NDPLVMs, which traditionally consider u as the sole inference network input. Our findings reveal that the key to achieving accurate inference of z lies in the label y, offering a significant insight that refines the conventional understanding in this domain.

4.4. Measure Change and Likelihood Term Approximation

Applying the Girsanov theorem, the posterior process under measure $\mathbb{Q}$ can be derived via the prior process under measure $\mathbb{P}$ as follows:

$$\mathrm{d}z^{\mathbb{Q}} = \exp(\int_{t-1}^{t} -\frac{1}{2}\|\nu\|^2 \mathrm{d}\tau)\mathrm{d}z^{\mathbb{P}}. \tag{26}$$

Since at start point $t-1, t \in [0, \mathrm{T})$, the probabilistic density of z under measure $\mathbb{Q}$ and $\mathbb{P}$ are same. The probabilistic density of z under measure $\mathbb{Q}$ at time t can be obtained via the probabilistic density of z at measure $\mathbb{P}$. Supposed the probabilistic density of z at time t is:

$$z_t \sim \mathcal{N}(\mu_t^p, \Sigma_t^p), \tag{27}$$

where superscript p indicates that z_t is in measure $\mathbb{P}$, μ and Σ are the mean and covariance of normal distributio, respectively. Suppose the solution of the integral of Radon-Nikodym derivative ν is:

$$\nu \sim \mathcal{N}(\mu_t^\nu, \Sigma_t^\nu). \tag{28}$$

Then the probabilistic density of z under measure $\mathbb{Q}$ can be derived as follow:

$$\begin{aligned} z \sim \mathcal{N}(&(\Sigma_t^\nu + \Sigma_t^p)^{-1}(\Sigma_t^\nu \mu_t^\nu + \Sigma_t^p \mu_t^p), \\ &(\Sigma_t^\nu + \Sigma_t^p)^{-1}(\Sigma_t^\nu \Sigma_t^p)), \end{aligned} \tag{29}$$

which can be regarded as Schur complement operation [13].

4.5. Derivation of Moment Expressions

In this subsection, the moment expressions approximation operations are proposed in detail to answer the question “Model Implementation within DL Backends".

4.5.1. Moment Expressions for Transition

According to reference [5], the distribution of an Itô process in time-axis can be represented by normal distribution denoted as $\mathcal{N}$, with a mean of μ and a covariance of Σ. Based on this, the following proposition for mean and covariance are given:

Proposition 4.3. *The mean and covariance equations between z_t and z_{t+1} can be given in (30) and (31), respectively:*

$$\frac{\mathrm{d}\mu}{\mathrm{d}t} = f(\mu, t), \tag{30}$$

$$\begin{aligned} \frac{\mathrm{d}\Sigma}{\mathrm{d}t} = &\Sigma[\frac{\partial f(z,t)}{\partial z}|_{z=\mu}]^\top \\ &+ \Sigma^\top[\frac{\partial f(z,t)}{\partial z}|_{z=\mu}] + L(\mu,t)QL^\top(\mu,t). \end{aligned} \tag{31}$$

Proof. The proof is provided in Section **??** of Supplementary Material. □

4.5.2. Moment Expressions for Loss Function

Based on previous subsubsection, we further derive the moment expressions of $\mathbb{E}_{\mathbb{Q}}\left[\|y_t - g(z_t, u_{1:t})\|^2\right]$ for inferential sensor task by following proposition:

Proposition 4.4. *The moment expressions of $\mathbb{E}_{\mathbb{Q}}[\|y_t - g(z_t, u_{1:t})\|^2]$ can be approximated as follow:*

$$\begin{aligned} &\mathbb{E}_{\mathbb{Q}}\left[\|y_t - g(z_t, u_{1:t})\|^2\right] \\ \approx&\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathcal{I})}[\mathcal{L}(\mu_{z,t}) + \frac{\partial \mathcal{L}}{\partial z}|_{z=\mu_{z,t}}\sigma_{z,t}(\frac{\partial \mathcal{L}}{\partial z}|_{z=\mu_{z,t}})^\top \times \epsilon] \\ :=&\mathcal{L}(\mu_{z,t}). \end{aligned} \tag{32}$$

Proof. The proof is given in Section **??** of Supplementary Material. □

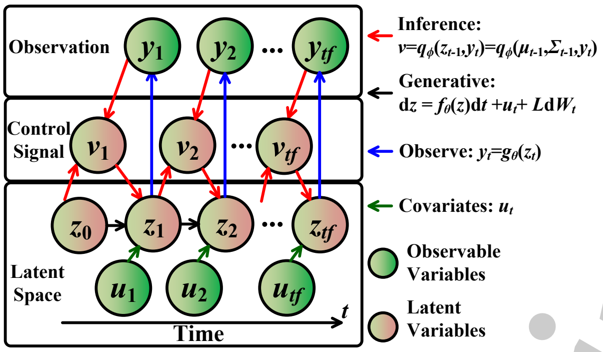

Figure 1 | The overall architecture of OC-NDPLVM

4.6. Model Overall Structure

Based on the above-mentioned subsections, the model architecture is summarized in Fig. 1. It can be seen that the model consists of two different colored nodes, where the green node represents the observable variables, and the orange node represents the latent variables. Note that the control policy is simulated by the inference network, where we assume that the neural network can infer the optimal control signal based on its input as assumed in (25). Furthermore, if we detract the control signal part in the middle of Fig. 1, the model degrades to current NDPLVMs for inferential sensor tasks. The detailed analysis of this Figure 1 and corresponding algorithm are given in Section **??** and **??** of Supplementary Material, respectively.

4.7. Theoretical Analysis of Learning Objective Convergence

In this subsection, we want to analyze the convergence of the proposed ADMM-based algorithm to make our work more complete. The following proposition is given for the convergence of proposed ADMM-based algorithm:

Proposition 4.5. *The convergence of the optimization procedure in* (11) *can be guaranteed, given that: 1), the inference network q_ϕ can simulate the optimal control signal ν^*; and 2) the learning rate in the stochastic gradient optimizer ensures the reduction of the loss function $\mathcal{L}$.*

Proof. The proof is given in Section **??** of Supplementary Material. □

Assumption 1) is a widely admitted setting in the context of “amortization variational inference"-based models as we mentioned in Section 3.1, and assumption 2) can be realized easily when the learning rate is low enough (analysis about this condition is given in Section **??** of Supplementary Material).

5. Experimental Results & Discussions

In this section, we devise experiments on two industrial inferential sensor datasets to verify the superiority of OC-NDPLVM and answer the research questions as follows:

- **Performance:** *Does OC-NDPLVMs work?* Section 5.2 evaluates OC-NDPLVM’s performance against a variety of baseline methods using two datasets from real industrial scenario, thereby establishing a foundational understanding of its operational efficiency.

- **Convergence:** *Does it converge?* To backup our theoretical analysis in Section 4.7. Section 5.3 analyzes the iteration curve for two datasets along epoch, thereby illustrating the convergence trajectory at the epoch scale.
- **Sensitivity:** *Is it sensitive to key hyperparameters?* Section **??** (Supplementary Material) elucidates the impact of different hyperparameters on the prediction accuracy, analyzing the system's responsiveness to parameter alterations.

5.1. Experimental Settings

Table 1 | The model performance on Inferential Sensor Task

Dataset	H	OC-NDPLVM		Informer		LogTrans		DPSFA		DPM		AR-TCN		DA-LSTM		DPMM		DMVAER	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
DB	2	**0.0141**	**0.0132**	0.0243†	0.0238†	0.0326†	0.0319†	0.1424†	0.1412†	0.1512†	0.1499†	0.0234†	0.0230†	0.0341†	0.0323†	0.1891†	0.1888†	0.1391†	0.1387†
	3	**0.0200**	**0.0181**	0.0314†	0.0305†	0.0344†	0.0333†	0.1413†	0.1394†	0.1433†	0.1419†	0.0298†	0.0292†	0.0393†	0.0367†	0.1895†	0.1889†	0.1395†	0.1387†
	4	**0.0260**	**0.0232**	0.0318†	0.0302†	0.0346†	0.0328†	0.1420†	0.1400†	0.1427†	0.1411†	0.0367†	0.0356†	0.0379†	0.0357†	0.1900†	0.1889†	0.1400†	0.1387†
	5	**0.0322**	**0.0284**	0.0351	0.0329†	0.0409†	0.0387†	0.1445†	0.1415†	0.1413†	0.1393†	0.0442†	0.0427†	0.0502†	0.048†	0.1904†	0.1890†	0.1405†	0.1387†
CSC	2	**0.1031**	**0.0939**	0.1071†	0.0978†	0.1065	0.0974	0.1450†	0.1329†	0.1309†	0.1204†	0.1036	0.0945	0.1793†	0.1631†	0.2118†	0.2116†	0.5285†	0.5090†
	3	**0.1081**	**0.0951**	0.1152	0.1023	0.1136	0.1008	0.1648†	0.1441†	0.1460†	0.1298†	0.1119	0.0991†	0.1965†	0.1713†	0.2121†	0.2116†	0.5377†	0.5092†
	4	**0.1148**	**0.0995**	0.1184	0.1033	0.1186	0.1036	0.1818†	0.1545†	0.1595†	0.1401†	0.1157	0.1007	0.2153†	0.1836†	0.2124†	0.2117†	0.5428†	0.5088†
	5	0.1199	0.1034	0.1218	0.1054	**0.1197**	**0.1033**	0.1948†	0.1612†	0.1563†	0.1350†	0.1201	0.1037	0.2268†	0.1910†	0.2127†	0.2117†	0.5427†	0.4972†

† marks the variants that OC-NDPLVM outperforms significantly at p-value < 0.05 over paired samples t-test. **Bolded** results indicate the best in each metric. Underlined results indicate the second best in each metric.

5.1.1. *Datasets*

We select two datasets namely debutanizer column (DC) and catalytic shift conversion (CSC). More details about these two datasets are given in Section **??** of Supplementary Material.

5.1.2. *Baseline Models*

We compare the proposed OC-NDPLVM with the following baseline models:

- Recurrent Network models: AR-TCN [14] and DA-LSTM [15];
- NDPLVMs: PDTM [16] and DBPSFA (state-of-the-art, 2023) [12];
- Self-attentive methods (non-auto-regressive structure): LogTrans [17] and Informer (state-of-the-art, 2021) [18].
- Mixture Model-based methods: Dirichlet process mixture model (DPMM) and dynamical mixture variational autoencoder regression (DMVAER) [19].

The reasons for choosing these models and other experimental details for experiments are also provided in Section **??** of Supplementary Material.

5.1.3. *Evaluation Metrics*

The RMSE and MAE are adopted as evaluation metrics. Their expressions are given as follows:

$$\mathrm{RMSE} = \frac{1}{\mathrm{N}}\frac{1}{\mathrm{H}}\sum_{n=1}^{\mathrm{N}}\sum_{h=1}^{\mathrm{N}}\sqrt{\left(\hat{y}_{h,n} - y_{h,n}\right)^2}, \tag{33}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\frac{1}{\mathrm{H}}\sum_{n=1}^{\mathrm{N}}\sum_{h=1}^{\mathrm{H}}\left|\hat{y}_{h,n} - y_{h,n}\right|, \tag{34}$$

where $\hat{y}$ represents the predicted values, y denotes the actual values, H is the length of the prediction horizon, and N signifies the total number of evaluation instances. For metrics such as RMSE and MAE, a lower value correlates with a more accurate model. Our evaluation metrics operate on a rolling basis along the time axis with a size-H prediction horizon. Consequently, in line with references [7, 18], we do not provide the conventional “prediction-real results" comparison graphs in our experimental results.

5.2. Overall Performance

In this subsection, question “*Does OC-NDPLVMs work?*" about performance comparison is answered. The comparison results for the OC-NDPLVM and other baseline models on the DC and CSC datasets are reported in Table 1. The following observations can be obtained:

1. For the DC dataset, the RMSE for $\mathrm{H} = 2, 3, 4, 5$ are 39.73% ~ 92.54%, 33.02% ~ 89.46%, 17.97% ~ 86.29%, and 8.14% ~ 83.08% lower than those of the baseline models, respectively; the MAE for $\mathrm{H} = 2, 3, 4, 5$ are 42.68% ~93.00%, 37.96% ~ 90.40%, 23.14% ~ 87.71%, and 13.72% ~ 84.97% lower than those of the baseline models, respectively.
2. For the CSC dataset, the RMSE for $\mathrm{H} = 2, 3, 4, 5$ are 0.45% ~ 80.49%, 3.43% ~ 79.90%, 0.79% ~ 78.86%, and 0.17% ~ 77.92% lower than those of the baseline models, respectively; the MAE for $\mathrm{H} = 2, 3, 4, 5$ are 0.65% ~ 81.55%, 4.06% ~ 81.32%, 1.21% ~ 80.45%, and 0.28% ~ 79.21% lower than those of the baseline models, respectively.
3. The performance gains of OC-NDPLVM against most baselines are significant, as is evidenced by the p-value< 0.05 over the paired samples t-test.
4. When the prediction window increases, the performance degradation of self-attention models is smaller than that of NDPLVMs and Recurrent models.

Observations 1) and 2) indicate that the proposed OC-NDPLVM outperforms other baseline models. Observation 3) indicates that it is sufficient to say the OC-NDPLVM is better than other baseline models for most of the scenarios. Observation 4) indicates that the models with auto-regressive structure may suffer from gradient vanishing with the increase of forecasting horizon, while the self-attention models can alleviate this issue thanks to their non-auto-regressive structure.

5.3. Convergence of the ADMM Framework

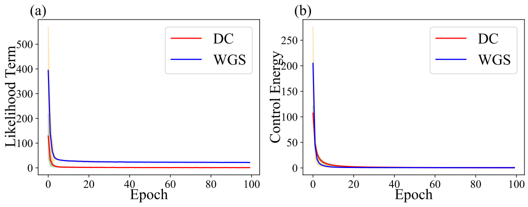

Figure 2 | Convergence analysis of the DC and CSC dataset for $\mathrm{H} = 4$, the (a) likelihood term ($\frac{1}{\mathrm{N}} \sum_{t=1}^{\mathrm{T+H}} \mathbb{E}_{\mathbb{Q}}[\log p(y_t|x_t, u_t)]$) and the (b) control energy term ($\frac{1}{\mathrm{N}} \sum_{t=1}^{\mathrm{T+H}} \mathbb{E}_{\mathbb{Q}}[\int_{t-1}^{t} \|u_\tau\|^2 d\tau]$) along iteration process. The shaded area indicates the ± 1.5 sigma uncertainty interval

In this subsection, question “*Does it converge?*" about convergence analysis is answered to practically demonstrate the ADMM optimization framework’s convergence. Fig. 2 (a) and (b) propose the likelihood term and the control energy term along the iteration process. From Fig. 2, it can be seen that both the likelihood term and energy term decrease with the increase of the iteration epoch. The decrease of the likelihood term along the training epoch in Fig. 2 (a) indicates that the latent state can represent the label pattern in the training process. Consequently, the decrease of the control energy with the increase of the training epoch in Fig. 2 (b) indicates that the dependence on the label information decreases in the training process. Specifically, the likelihood and control energy terms tend to be unchanged after 10 epochs. This phenomenon reflects that the optimization strategy built upon ADMM framework has a fast convergence rate. Furthermore, it can be observed that the energy of the control signal tends to approach zero after a few epochs. This suggests that the approximation utilized in the assumption in the proof of Proposition 4.4 (Section **??**, Supplementary Material) is reasonable.

6. Conclusions

In this work, to answer two fundamental but essential problems, namely “inaccurate inference of latent space" and “model implementation within DL backends" in the NDPLVMs, we first proposed our OC-NDPLVM and its loss function from SDE theory, conducted detailed analysis of model training algorithm, derived moment expressions, and summarized the model overall architecture. In this procedure, we redesigned the input of inference network by solving the optimal control subproblem and simplified the model implementation by obtaining the moment expressions. Finally, to empirically validate the proposed method's effectiveness, we conduct various inferential sensor experiments on two industrial datasets.

We think future directions can be focused on the introducing other integrators to alleviate the stiff issue of ODEs. Besides, other distribution transformation methods like the sequential Monte-Carlo method can also be adopted to estimate the likelihood function more accurately.

References

[1] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, volume 32, pages 1–12, 2019.

[2] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in variational inference. IEEE Transactions on Pattern Analysis & Machine Intelligence, 41(08):2008–2026, aug 2019. ISSN 1939-3539.

[3] Ankush Ganguly, Sanjana Jain, and Ukrit Watchareeruetai. Amortized variational inference: A systematic review. Journal of Artificial Intelligence Research, 78:167–215, 2023.

[4] Francisco Vargas. Machine-learning approaches for the empirical schrödinger bridge problem. Technical report, University of Cambridge, Computer Laboratory, 2021.

[5] Simo Särkkä and Arno Solin. Applied stochastic differential equations, volume 10. Cambridge University Press, 2019.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers, volume 3. Now Publishers, Inc., 2011.

[7] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. pages 6778–6786, 8 2023. doi: 10.24963/ijcai.2023/759. URL https://doi.org/10.24963/ijcai.2023/759. Survey Track.

[8] Dimitri Bertsekas. A Course in Reinforcement Learning. Athena Scientific, 2023.

[9] Lev Semenovich Pontryagin, EF Mishchenko, VG Boltyanskii, and RV Gamkrelidze. The mathematical theory of optimal processes. 1962.

[10] Hector O Fattorini. Infinite dimensional optimization and control theory, volume 54. Cambridge University Press, 1999.

[11] Bingbing Shen and Zhiqiang Ge. Supervised nonlinear dynamic system for soft sensor application aided by variational auto-encoder. IEEE Transactions on Instrumentation and Measurement, 69(9):6132–6142, 2020. doi: 10.1109/TIM.2020.2968162.

[12] Chao Jiang, Yusheng Lu, Weimin Zhong, Biao Huang, Dayu Tan, Wenjiang Song, and Feng Qian. Deep bayesian slow feature extraction with application to industrial inferential modeling. IEEE Transactions on Industrial Informatics, 19(1):40–51, 2023. doi: 10.1109/TII.2021.3129888.

[13] Carl Edward Rasmussen. Gaussian processes in machine learning. In Summer school on machine learning, pages 63–71. Springer, 2003.

[14] Xiaofeng Yuan, Shuaibin Qi, Yalin Wang, Kai Wang, Chunhua Yang, and Lingjian Ye. Quality variable prediction for nonlinear dynamic industrial processes based on temporal convolutional networks. IEEE Sensors Journal, 21(18):20493–20503, 2021. doi: 10.1109/JSEN.2021.3096215.

[15] Liangjun Feng, Chunhui Zhao, and Youxian Sun. Dual attention-based encoder–decoder: A customized sequence-to-sequence learning for soft sensor development. IEEE Transactions on Neural Networks and Learning Systems, 32(8):3306–3317, 2021. doi: 10.1109/TNNLS.2020.3015929.

[16] Yusheng Lu, Xin Peng, Dan Yang, Chao Jiang, and Weimin Zhong. The probabilistic discriminative time-series model with latent variables and its application to industrial chemical process modeling. Chemical Engineering Journal, 423:130298, 2021. ISSN 1385-8947.

[17] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in Neural Information Processing Systems (NeuralIPS 19), 32, 2019.

[18] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, volume 35, pages 11106–11115, 2021.

[19] Le Yao, Bingbing Shen, Linlin Cui, Junhua Zheng, and Zhiqiang Ge. Semi-supervised deep dynamic probabilistic latent variable model for multimode process soft sensor application. IEEE Transactions on Industrial Informatics, 19(4):6056–6068, 2023. doi: 10.1109/TII.2022.3183211.